# Shortest path algorithms using DP

Directed $G=(V,E)$ with edge weights $w(e)$

**Fix $s \in V$:**

**Dijkstra's** algorithm finds for all $z \in V$

$dist(z)$ = length of shortest path from s to z

in time $O((n+m)\log n)$ $\quad n=|V|$, $m=|E|$

assuming all edge weights are positive.

What if **negative** weight edges?

May be negative weight cycles

example:


What is $dist(d)$?

Assume no negative weight cycles for now.
(see how to deal with them later.)

Solve using DP.

Note: for shortest path $P$ from $s$ to $z$, $P$ visits every vertex at most once if no negative weight cycle.

So $P$ is of length $\leq n-1$ edges.

DP idea: use $i = 0 \to n-1$ edges on the path.

For $0 \leq i \leq n-1$, and $z \in V$,
let $D(i,z)$ = length of shortest path from $s$ to $z$ using $\leq i$ edges.

Base case: $D(0,s) = 0$
for $z \neq s$, $D(0,z) = \infty$

For $i \geq 1$: look at a shortest path $P$ from $s$ to $z$ of length $i$


last edge goes from some $y$ to $z$

this is a shortest path from $s$ to $y$ using $\leq i-1$ edges

Thus, $D(i,z) = \min_{y} \left\{ D(i-1,y) + w(y,z) \right\}$

$\text{Min}\{D(i-1,z),$

**Bellman-Ford$(G,s,w)$:**

for all $z \in V$, $D(0,z) = \infty$

$D(0,s) = 0$

for $i = 1 \to n-1$

  for all $z \in V$, $D(i,z) = \cancel{\infty}\, D(i-1,z)$

  for all $(y,z) \in E$:

    if $D(i,z) > D(i-1,y) + w(y,z)$

      then $D(i,z) = D(i-1,y) + w(y,z)$

Running time: $O(nm)$.

How to find a negative weight cycle?

Check if ~~$D(n-1, z) < 0$~~ for some $z \in V$.

$D(n,z) < D(n-1,z)$

Now what if we want all-pairs shortest paths?

Then if we run Bellman-Ford $n$ times, we get a $O(n^2 m)$ time algorithm.

Better approach: Use Dynamic Programming

Let $dist(v,z)$ = length of shortest path from $v$ to $z$

Do we condition on # of edges as in Bellman-Ford?

$\Rightarrow$ This gives a $O(n^2 m)$ time algorithm.

Instead condition on the set of intermediate vertices.

Order the vertices: so $V = \{v_1, v_2, \ldots, v_n\}$

For $0 \le i \le n$ & $v, z \in V$,

let $D(i, v, z)$ = length of shortest path from $v$ to $z$ using a subset of $\{v_1, \dots, v_i\}$ as intermediate vertices

Base case:

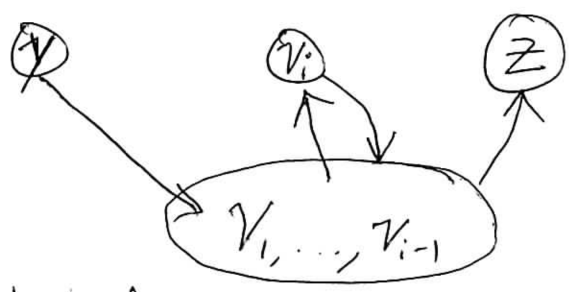
$$D(0, v, z) = \begin{cases} w(v, z) & \text{if } (v, z) \in E \\ \infty & \text{if } (v, z) \notin E \end{cases}$$

How to write a recurrence?

Look at $D(i, v, z)$ for $i \ge 1$:

if $v_i$ is not used then: $D(i, v, z) = D(i-1, v, z)$

if $v_i$ is used then the path looks like:




$$\underbrace{v - \text{subset of } \{v_1, \dots, v_{i-1}\} - v_i}_{\text{length is } D(i-1, v, v_i)} - \underbrace{\text{subset of } \{v_1, \dots, v_{i-1}\} - z}_{\text{length } D(i-1, v_{i-1}, z)}$$

— Assuming no negative weight cycle

Hence for $i \geq 1$ & vertices $y, z \in V$:

$$D(i, y, z) = \min\{D(i-1, y, z), D(i-1, y, v_i) + D(i-1, v_i, z)\}$$

Floyd-Warshall$(G, w)$:

For all $y \in V$:
  For all $z \in V$:
    if $(y, z) \in E$
      then $D(0, y, z) = w(y, z)$
      else $D(0, y, z) = \infty$

For $i = 1 \rightarrow n$
  For all $y \in V$:
    For all $z \in V$:
      $D(i, y, z) = \min\{D(i-1, y, v_i) + D(i-1, v_i, z), D(i-1, y, z)\}$

Return$(D(n, \cdot, \cdot))$

— Running time: $O(n^3)$

How do we check for a negative-weight cycle?

Check if $D(n,v,v) < 0$ for some $v \in V$.